A high-order accurate discretization scheme for variable coefficient elliptic PDEs in the plane with smooth solutions

P.G. Martinsson, Department of Applied Mathematics, University of Colorado at Boulder

Abstract: A discretization scheme for variable coefficient elliptic PDEs in the plane is presented. The scheme is based on high-order Gaussian quadratures and is designed for problems with smooth solutions, such as scattering problems involving soft scatterers. The resulting system of linear equations is very well suited to efficient direct solvers such as nested dissection and the more recently proposed accelerated nested dissection schemes with $O(N)$ complexity.

1. Introduction

1.1. **Background.** This note describes some tentative ideas for how to discretize and solve a class of variable-coefficient elliptic PDEs with smooth solutions. The ultimate goal is to device efficient methods for (soft) scattering problems in the plane, modeled by the Helmholtz' equation

$$-\Delta\,\phi(x) - \frac{\omega^2}{c(x)^2}\,\phi(x) = 0, \qquad x \in \mathbb{R}^2,$$

where $c(x)$ is a smooth function that is constant outside some domain Ω, and the "loading" of the system is an incoming wave that satisfies the Helmholtz equation for the constant value of c outside Ω.

The method described is high-order accurate and and leads to a linear system of algebraic equations that is very well-suited to "nested-dissection" type *direct* (as opposed to *iterative*) solvers. In a simple implementation, the resulting solver has $O(N^{1.5})$ complexity, where N is the total number of degrees of freedom in the discretization. We believe that the cost of the nested dissection step can be further reduced to $O(N)$ by exploiting techniques similar to those of [1, 2, 3].

In this initial work, we make several simplifying assumptions in order to investigate the basic viability of the method. The most significant simplification is that instead of studying the Helmholtz equation (which has oscillatory solutions), we study the modified Helmholtz equation (which has non-oscillatory solutions the decay exponentially fast). The remaining simplifications are, we believe, mostly cosmetic.

1.2. **Problem statement.** We consider the equation

$$\begin{cases} -\nabla(a(x)\nabla\phi(x)) + b(x)\,\phi(x) = & 0, & x \in \Omega, \\ \phi_n(x) = & v(x), & x \in \Gamma, \end{cases} \tag{1.1}$$

where Ω is a box in the plane with boundary $\Gamma = \partial\Omega$, and where ϕ_n is the normal derivative of ϕ. We assume that the functions a and b are C^∞, that $b \geq 0$, and that for every $x \in \overline{\Omega}$, $a(x)$ is positive definite. Under these assumptions, the solution ϕ and its gradient $\nabla\phi$ will be C^∞ in the interior of the domain and can to very high accuracy be specified via tabulation at Gaussian quadrature nodes. Near the boundary Γ, the question of smoothness in general gets complicated, but in this preliminary report we sidestep this issue by assuming that the given boundary data v is such that the solution ϕ is C^∞ on the closed domain $\overline{\Omega}$. (Ultimately, the technique will be applied to scattering problems and the function v will be the restriction to Γ of the "incoming wave.")

1.3. **Outline of the discretization scheme.** We propose to tessellate the computational domain into a large number of small squares and then use the fluxes across the boundaries of each square as the unknown variables in the model. The flux is represented as a function along the edge; since this function is smooth, it can very accurately be represented by simply tabulating it at Gaussian nodes along the edge. We observe that if the fluxes through all four edges of a box are known, then

the values of the potential ϕ on all of the edges can be constructed via the so called "Neumann-to-Dirichlet" (N2D) operator for the box. These operators can cheaply be constructed for all the boxes via a local computation. Once the N2D operators for all boxes are known, we construct for each edge an equilibrium equation by combining the N2D operators of the two boxes that share the edge. By combining the equilibrium equations for all interior edges, we obtain a global equation for all the interior boundary fluxes. Once this global equation has been solved, the potential on any box can easily be reconstructed by solving a local Neumann problem on the box (since the boundary fluxes are now all known).

1.4. **Outline of the linear solver.** The discretization described in Section 1.3 results in a large sparse linear system with a coefficient matrix A. If there are N_{edge} edges in the model, and we place N_{gauss} interpolation nodes at each edge, then A is a block matrix consisting of $N_{\text{edge}} \times N_{\text{edge}}$ blocks, each of size $N_{\text{gauss}} \times N_{\text{gauss}}$. By ordering the edges in a nested dissection fashion, fill-in can be limited in the factorization of A, resulting in an $O(N_{\text{edge}}^{1.5})$ total cost for the initial solve. (Once one factorization has been executed, subsequent solves require $O(N_{\text{edge}})$ operations.)

The dominant cost in the factorization described above is the inversion or factorization of dense matrices of size roughly $N_{\text{edge}}^{1/2} \times N_{\text{edge}}^{1/2}$. These matrices have internal structure (they are so call *Hierarchically Semi-Separable* (HSS) matrices) which can be exploited to further reduce the complexity to $O(N_{\text{edge}})$.

2. Discretization

We tessellate Ω into an array of small boxes, and let $\{\Gamma^{(i)}\}_{i \in I_{\text{leaves}}}$ denote the collection of *edges* of these boxes, see Figure 1. We include both interior and exterior edges. For an edge i, we define $u^{(i)}$ as the restriction of ϕ to $\Gamma^{(i)}$:

$$u^{(i)}(x) = \phi(x), \qquad \text{for } x \in \Gamma^{(i)}.$$

Further, we define $v^{(i)}$ as the restriction of the normal derivative across $\Gamma^{(i)}$:

$$v^{(i)}(x) = \begin{cases} [\partial_2 \phi](x) & \text{for } x \in \Gamma^{(i)} \text{ when } \Gamma^{(i)} \text{ is horizontal,} \\ [\partial_1 \phi](x) & \text{for } x \in \Gamma^{(i)} \text{ when } \Gamma^{(i)} \text{ is vertical,} \end{cases}$$

where we used the short-hand $\partial_i = \partial/\partial x_i$.

Observation: All the functions $u^{(i)}$ and $v^{(i)}$ are smooth. They can to very high accuracy be specified by tabulating them at Gaussian points on the boundary and then interpolate between these points.

On each line $\Gamma^{(i)}$, we place N_{gauss} Gaussian nodes. These points are collected in vectors $\boldsymbol{\gamma}^{(i)} \in \mathbb{R}^{N_{\text{gauss}} \times 2}$. Then we form vectors $\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(i)} \in \mathbb{R}^{N_{\text{gauss}}}$ by collocating the boundary functions $u^{(i)}$ and $v^{(i)}$ at the Gaussian nodes:

$$\boldsymbol{u}^{(i)} = u^{(i)}(\boldsymbol{\gamma}^{(i)}),$$
$$\boldsymbol{v}^{(i)} = v^{(i)}(\boldsymbol{\gamma}^{(i)}).$$

3. The equilibrium equations

3.1. **Definition of the Neumann-to-Dirichlet operator.** Let $\Omega^{(\tau)}$ be a subdomain of Ω with edges $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$, as shown in Figure 2. We define the boundary potentials and boundary fluxes for $\Omega^{(\tau)}$ via

$$u^{(\tau)} = \begin{bmatrix} u^{(i_1)} \\ u^{(i_2)} \\ u^{(i_3)} \\ u^{(i_4)} \end{bmatrix} \qquad \text{and} \qquad v^{(\tau)} = \begin{bmatrix} v^{(i_1)} \\ v^{(i_2)} \\ v^{(i_3)} \\ v^{(i_4)} \end{bmatrix}.$$

Then there exists a unique operator $T^{(\tau)}$ such that with $u^{(\tau)}$ and $v^{(\tau)}$ derived from any solution ϕ to (1.1), we have

$$u^{(\tau)} = T^{(\tau)}\, v^{(\tau)}. \tag{3.1}$$

This claim follows from the fact that a Neumann boundary value problem on $\Omega^{(\tau)}$ has a unique solution. The operator $T^{(\tau)}$ is mathematically an integral operator called the *Neumann-to-Dirichlet* operator.

Observation: The N2D operator is in general a complicated object. It has a singular kernel even for domains with smooth boundaries. When the domain boundary has corners, further complications arise. However, in our case, all such subtleties can be ignored since the boundary potentials of interest are all restrictions of functions that globally solve (1.1), and are in consequence smooth.

The discrete analog of the equation (3.1) is

$$\boldsymbol{u}^{(\tau)} = \mathsf{T}^{(\tau)}\, \boldsymbol{v}^{(\tau)}. \tag{3.2}$$

For our purposes, it is sufficient for the matrix $\mathsf{T}^{(\tau)}$ to correctly construct $\boldsymbol{u}^{(\tau)}$ for any *permissible* vectors $\boldsymbol{v}^{(\tau)}$. (By permissible, we mean that they are the restriction of a function in the solution set under consideration.) Written out in components, $\mathsf{T}^{(\tau)}$ is a 4×4 block matrix that satisfies

$$\begin{bmatrix} \boldsymbol{u}^{(i_1)} \\ \boldsymbol{u}^{(i_2)} \\ \boldsymbol{u}^{(i_3)} \\ \boldsymbol{u}^{(i_4)} \end{bmatrix} = \begin{bmatrix} \mathsf{T}^{(\tau,11)} & \mathsf{T}^{(\tau,12)} & \mathsf{T}^{(\tau,13)} & \mathsf{T}^{(\tau,14)} \\ \mathsf{T}^{(\tau,21)} & \mathsf{T}^{(\tau,22)} & \mathsf{T}^{(\tau,23)} & \mathsf{T}^{(\tau,24)} \\ \mathsf{T}^{(\tau,31)} & \mathsf{T}^{(\tau,32)} & \mathsf{T}^{(\tau,33)} & \mathsf{T}^{(\tau,34)} \\ \mathsf{T}^{(\tau,41)} & \mathsf{T}^{(\tau,42)} & \mathsf{T}^{(\tau,43)} & \mathsf{T}^{(\tau,44)} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{(i_1)} \\ \boldsymbol{v}^{(i_2)} \\ \boldsymbol{v}^{(i_3)} \\ \boldsymbol{v}^{(i_4)} \end{bmatrix}. \tag{3.3}$$

3.2. **Construction of the N2D operator for a small box.** Let $\Omega^{(\tau)}$ be a small box with edges $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$, as shown in Figure 2, and consider the task of constructing a matrix $\mathsf{T}^{(\tau)}$ such that (3.2) holds for all permissible potentials. We generate (by brute force) a collection of solutions $\{\phi_j\}_{j=1}^{N_{\text{samp}}}$ that locally span the solution space to the desired precision. For each ϕ_j, we construct the corresponding vectors of boundary values

$$\boldsymbol{u}_j = \begin{bmatrix} \phi_j(\boldsymbol{\gamma}^{(i_1)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_2)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_3)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_4)}) \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{v}_j = \begin{bmatrix} \partial_2\phi_j(\boldsymbol{\gamma}^{(i_1)}) \\ \partial_1\phi_j(\boldsymbol{\gamma}^{(i_2)}) \\ \partial_2\phi_j(\boldsymbol{\gamma}^{(i_3)}) \\ \partial_1\phi_j(\boldsymbol{\gamma}^{(i_4)}) \end{bmatrix}$$

and then we construct via a least squares procedure a matrix $\mathsf{T}^{(\tau)}$ such that the equation

$$[\boldsymbol{u}_1\ \boldsymbol{u}_2\ \cdots\ \boldsymbol{u}_{N_{\text{samp}}}] = \mathsf{T}^{(\tau)}\, [\boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_{N_{\text{samp}}}] \tag{3.4}$$

holds to within the specified tolerance ε.

The sample functions ϕ_j are constructed by solving a set of local problems on a patch Ψ that covers the domain $\Omega^{(\tau)}$, as shown in Figure 5. The local problems read

$$\begin{cases} -\nabla(\tilde{a}(x)\nabla\phi_j(x)) + \tilde{b}(x)\,\phi_j(x) = & 0, & x \in \Psi, \\ \partial_n\phi(x) = & v_j(x), & x \in \partial\Psi, \end{cases} \tag{3.5}$$

where $\tilde{a}$ and $\tilde{b}$ are functions chosen so that:

(1) For $x \in \Omega^{(\tau)}$, we have $\tilde{a}(x) = a(x)$ and $\tilde{b}(x) = b(x)$.
(2) The equation (3.5) is easy to solve.

The collection of boundary data $\{v_j\}_{j=1}^{N_{\text{samp}}}$ is chosen so that the solution space is sufficiently "rich."

3.3. **Assembling a global equilibrium equation.** In this section, we will formulate a linear equation that relates the following variables:

$$\text{Given data:} \quad \{\boldsymbol{v}^{(i)} : i \text{ is an edge that is exterior to } \Omega\},$$
$$\text{Sought data:} \quad \{\boldsymbol{v}^{(i)} : i \text{ is an edge that is interior to } \Omega\}.$$

Let N_{edge} denote the number of interior edges. Then the coefficient matrix of the linear system will consist of $N_{\text{edge}} \times N_{\text{edge}}$ blocks, each of size $N_{\text{gauss}} \times N_{\text{gauss}}$. Each block row in the system will have at most 7 non-zero blocks. To form this matrix, let i denote an interior edge. Suppose for a moment that i is a vertical edge. Let τ_1 and τ_2 denote the two boxes that share the edge i, let $\{m_1, m_2, m_3, m_4\}$ denote the edges of τ_1, and let $\{n_1, n_2, n_3, n_4\}$ denote the edges of τ_2, see Figure 3. The N2D operator for τ_1 provides an equation for the boundary fluxes of the left box:

$$\boldsymbol{u}^{(m_2)} = \mathsf{T}^{(\tau_1,21)}\,\boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\tau_1,22)}\,\boldsymbol{v}^{(m_2)} + \mathsf{T}^{(\tau_1,23)}\,\boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\tau_1,24)}\,\boldsymbol{v}^{(m_4)}. \tag{3.6}$$

Analogously, the N2D operator for τ_2 provides the equation

$$\boldsymbol{u}^{(n_4)} = \mathsf{T}^{(\tau_2,41)}\,\boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\tau_2,42)}\,\boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\tau_2,43)}\,\boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\tau_2,44)}\,\boldsymbol{v}^{(n_4)}. \tag{3.7}$$

Observing that $m_2 = n_2 = i$, we see that $\boldsymbol{u}^{(m_2)} = \boldsymbol{u}^{(n_4)}$, and consequently (3.6) and (3.7) can be combined to form the equation

$$\begin{aligned}\mathsf{T}^{(\tau_1,21)}\,\boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\tau_1,22)}\,\boldsymbol{v}^{(i)} + \mathsf{T}^{(\tau_1,23)}\,\boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\tau_1,24)}\,\boldsymbol{v}^{(m_4)} \\ = \mathsf{T}^{(\tau_2,41)}\,\boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\tau_2,42)}\,\boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\tau_2,43)}\,\boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\tau_2,44)}\,\boldsymbol{v}^{(i)}.\end{aligned} \tag{3.8}$$

The collection of all equations of the form (3.8) for interior vertical edges, along with the analogous set of equations for all interior horizontal edges forms the global equilibrium equation.

4. Efficient direct solvers

This section describes a direct solver for the global equilibrium equation constructed in Section 3. The idea is to partition the box Ω into a quad-tree of boxes, and then to construct the N2D operator $\mathsf{T}^{(\tau)}$ for each box τ in the tree. The first step is to loop over all leaf nodes of the tree and construct the N2D operator via the procedure described in Section 3.2. Then we execute an upwards sweep through the tree, where we construct the N2D operator for a box by merging the operators of its four children. For simplicity (and also computational efficiency) we execute each "merge-four" operation as a set of three "merge-two" operations.

Let N denote the size of the coefficient matrix. Then Section 4.2 describes a procedure with $O(N^{1.5})$ complexity, and Section 4.3 sketches out how the procedure can be accelerated to $O(N)$ complexity. Before describing the fast solvers, we describe a hierarchical decomposition of the domain in Section 4.1.

4.1. **A quad-tree on the domain.** A standard quad-tree is formed on the computational domain Ω as follows: Let $\Omega^{(1)} = \Omega$ be the *root* of the tree, as shown in Figure 4(a). Then split $\Omega^{(1)}$ into four disjoint boxes

$$\Omega^{(1)} = \Omega^{(2)} \cup \Omega^{(3)} \cup \Omega^{(4)} \cup \Omega^{(5)},$$

as shown in Figure 4(b). Continue by splitting each of the four boxes into four smaller equisized boxes:

$$\begin{aligned}\Omega^{(5)} &= \Omega^{(6)} \cup \Omega^{(7)} \cup \Omega^{(8)} \cup \Omega^{(9)},\\ \Omega^{(6)} &= \Omega^{(10)} \cup \Omega^{(11)} \cup \Omega^{(12)} \cup \Omega^{(13)},\\ \Omega^{(7)} &= \Omega^{(14)} \cup \Omega^{(15)} \cup \Omega^{(16)} \cup \Omega^{(17)},\\ \Omega^{(8)} &= \Omega^{(18)} \cup \Omega^{(19)} \cup \Omega^{(20)} \cup \Omega^{(21)},\end{aligned}$$

as shown in Figure 4(c). The process continues until each box is small enough that the N2D operator for each leaf can easily be constructed via the procedure described in Section 3.2. The levels of the tree are ordered so that $\ell = 0$ is the coarsest level (consisting only of the root), $\ell = 1$ is the level with four boxes, etc. We let L denote the total number of levels in the tree.

4.2. **Simple construction of the N2D operator for a parent.** Suppose that σ is a box with children ν_1 and ν_3 as shown in Figure 6, and that we know the matrices $\mathsf{T}^{(\nu_1)}$ and $\mathsf{T}^{(\nu_3)}$ associated with the children. We seek to construct the matrix $\mathsf{T}^{(\sigma)}$. The equilibrium equations for the two children read

$$\boldsymbol{u}^{(m_i)} = \sum_{j=1}^{4} \mathsf{T}^{(\nu_1,ij)}\, \boldsymbol{v}^{(m_j)}, \qquad i = 1,\, 2,\, 3,\, 4, \tag{4.1}$$

$$\boldsymbol{u}^{(n_i)} = \sum_{j=1}^{4} \mathsf{T}^{(\nu_3,ij)}\, \boldsymbol{v}^{(n_j)}, \qquad i = 1,\, 2,\, 3,\, 4. \tag{4.2}$$

Observing that $\boldsymbol{u}^{(m_2)} = \boldsymbol{u}^{(n_n)}$ we combine (4.1) for $i = 2$ with (4.2) for $i = 4$ to obtain the joint equation

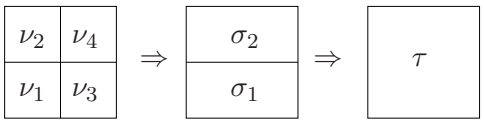
$$\begin{aligned}\mathsf{T}^{(\nu_1,21)}\, \boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\nu_1,22)}\, \boldsymbol{v}^{(m_2)} + \mathsf{T}^{(\nu_1,23)}\, \boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\nu_1,24)}\, \boldsymbol{v}^{(m_4)} \\ = \mathsf{T}^{(\nu_3,41)}\, \boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\nu_3,42)}\, \boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\nu_3,43)}\, \boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\nu_3,44)}\, \boldsymbol{v}^{(n_4)}.\end{aligned} \tag{4.3}$$

Utilizing further that $\boldsymbol{v}^{(m_2)} = \boldsymbol{v}^{(n_4)}$, we write (4.3) along with (4.1) and (4.2) as

$$\left[\begin{array}{ccc|ccc|c} \mathsf{T}^{(\nu_1,11)} & \mathsf{T}^{(\nu_1,13)} & \mathsf{T}^{(\nu_1,14)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,12)} \\ \mathsf{T}^{(\nu_1,31)} & \mathsf{T}^{(\nu_1,33)} & \mathsf{T}^{(\nu_1,34)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,32)} \\ \mathsf{T}^{(\nu_1,41)} & \mathsf{T}^{(\nu_1,43)} & \mathsf{T}^{(\nu_1,44)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,42)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,11)} & \mathsf{T}^{(\nu_3,12)} & \mathsf{T}^{(\nu_3,13)} & \mathsf{T}^{(\nu_3,14)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,21)} & \mathsf{T}^{(\nu_3,22)} & \mathsf{T}^{(\nu_3,23)} & \mathsf{T}^{(\nu_3,24)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,31)} & \mathsf{T}^{(\nu_3,32)} & \mathsf{T}^{(\nu_3,33)} & \mathsf{T}^{(\nu_3,34)} \\ \hline \mathsf{T}^{(\nu_1,21)} & \mathsf{T}^{(\nu_1,23)} & \mathsf{T}^{(\nu_1,24)} & -\mathsf{T}^{(\nu_3,41)} & -\mathsf{T}^{(\nu_3,42)} & -\mathsf{T}^{(\nu_3,43)} & \mathsf{T}^{(\nu_1,22)} - \mathsf{T}^{(\nu_3,44)} \end{array}\right] \left[\begin{array}{c} \boldsymbol{v}^{(m_1)} \\ \boldsymbol{v}^{(m_3)} \\ \boldsymbol{v}^{(m_4)} \\ \boldsymbol{v}^{(n_1)} \\ \boldsymbol{v}^{(n_2)} \\ \boldsymbol{v}^{(n_3)} \\ \hline \boldsymbol{v}^{(m_2)} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{u}^{(m_1)} \\ \boldsymbol{u}^{(m_3)} \\ \boldsymbol{u}^{(m_4)} \\ \boldsymbol{u}^{(n_1)} \\ \boldsymbol{u}^{(n_2)} \\ \boldsymbol{u}^{(n_3)} \\ \hline \boldsymbol{0} \end{array}\right].$$

Eliminating $\boldsymbol{v}^{(m_2)}$ from the system via a Schur complement yields the operator $\mathsf{T}^{(\sigma)}$ (upon suitable reblocking).

At this point, we have described a "merge-two" operation. A "merge-four" operation can of course be obtained by simply combining three merge-two operations. To be precise, suppose that τ is a node with the four children ν_1, ν_2, ν_3, ν_4. We introduce the two "intermediate" boxes σ_1 and σ_2 as shown in the following figure:



Letting the procedure described earlier in the section be denoted by "`merge_two_horizontal`" and defining an analogous function "`merge_two_vertical`" we then find that

$$\begin{aligned}\mathsf{T}^{(\sigma_1)} &= \texttt{merge\_two\_horizontal}(\mathsf{T}^{(\nu_1)},\, \mathsf{T}^{(\nu_3)}),\\ \mathsf{T}^{(\sigma_2)} &= \texttt{merge\_two\_horizontal}(\mathsf{T}^{(\nu_2)},\, \mathsf{T}^{(\nu_4)}),\\ \mathsf{T}^{(\tau)} &= \texttt{merge\_two\_vertical}(\mathsf{T}^{(\sigma_1)},\, \mathsf{T}^{(\sigma_2)}).\end{aligned}$$

4.3. **Fast construction of the N2D operator for a parent.** The merge operation described in Section 4.2 has asymptotic cost $O(N^{1.5})$, where N is the total number of points on the edges of the leaves. To simplify slightly, the reason is that forming the "merge" operation requires matrix inversion and matrix-matrix-multiplications for of dense matrices whose size eventually grow to $O(\sqrt{N}) \times O(\sqrt{N})$. However, these matrices all have internal structure. To be precise, in (3.3), the off-diagonal blocks have low rank, and the diagonal blocks are all Hierarchically Semi-Separable (HSS) matrices. This means that accelerated matrix algebra can be used. For a matrix of size

$N' \times N'$, inversion can in fact be executed in $O(N')$ operations (provided that the "HSS-rank" is a fixed low number, which it is in this case).

The acceleration procedure proposed here is analogous to the one described in [1, 2, 3].

References

[1] Sabine Le Borne, Lars Grasedyck, and Ronald Kriemann, *Domain-decomposition based $\mathcal{H}$-LU preconditioners*, Domain decomposition methods in science and engineering XVI, Lect. Notes Comput. Sci. Eng., vol. 55, Springer, Berlin, 2007, pp. 667–674. MR 2334161

[2] P. Schmitz and L. Ying, *A fast direct solver for elliptic problems on general meshes in 2d*, In review.

[3] Jianlin Xia, Shivkumar Chandrasekaran, Ming Gu, and Xiaoye S. Li, *Superfast multifrontal method for large structured linear systems of equations*, SIAM J. Matrix Anal. Appl. **31** (2009), no. 3, 1382–1411. MR 2587783

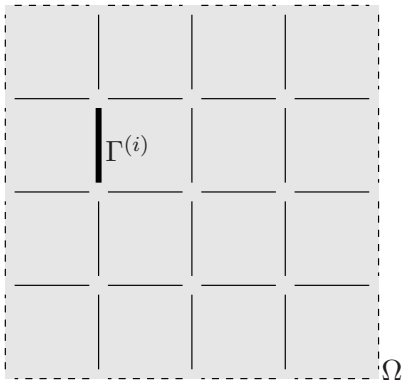


FIGURE 1. The computational box Ω (gray) is split into 16 small boxes. There are a total of 40 edges in the discretization, 24 interior ones (solid lines) and 16 exterior ones (dashed lines). One interior edge $\Gamma^{(i)}$ is marked with a bold line. (Each edge continues all the way to the corner, but has been drawn slightly forshortened for clarity.)

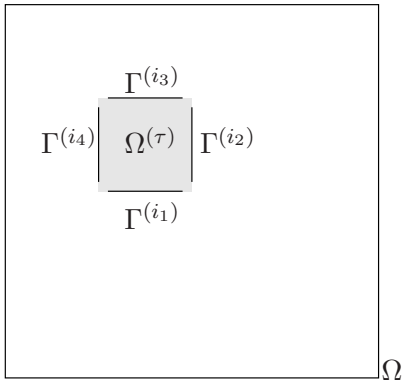


FIGURE 2. The box $\Omega^{(\tau)}$ is marked in gray. Its edges are $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$.

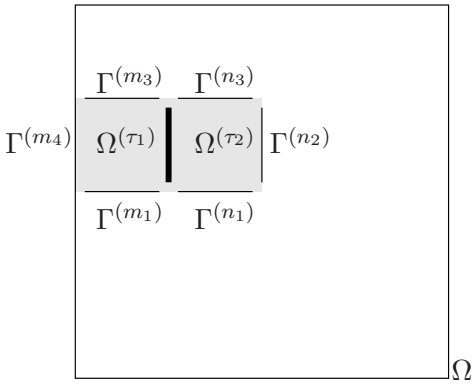


FIGURE 3. Construction of the equilibrium equation for the edge $\Gamma^{(i)}$ in Figure 1. It is the common edge of the boxes $\Omega^{(\tau_1)}$ and $\Omega^{(\tau_2)}$, which have edges $\{\Gamma^{(m_1)}, \Gamma^{(m_2)}, \Gamma^{(m_3)}, \Gamma^{(m_4)}\}$, and $\{\Gamma^{(n_1)}, \Gamma^{(n_2)}, \Gamma^{(n_3)}, \Gamma^{(n_4)}\}$, respectively. Observe that $\Gamma^{(i)} = \Gamma^{(m_2)} = \Gamma^{(n_4)}$ (the bold line).

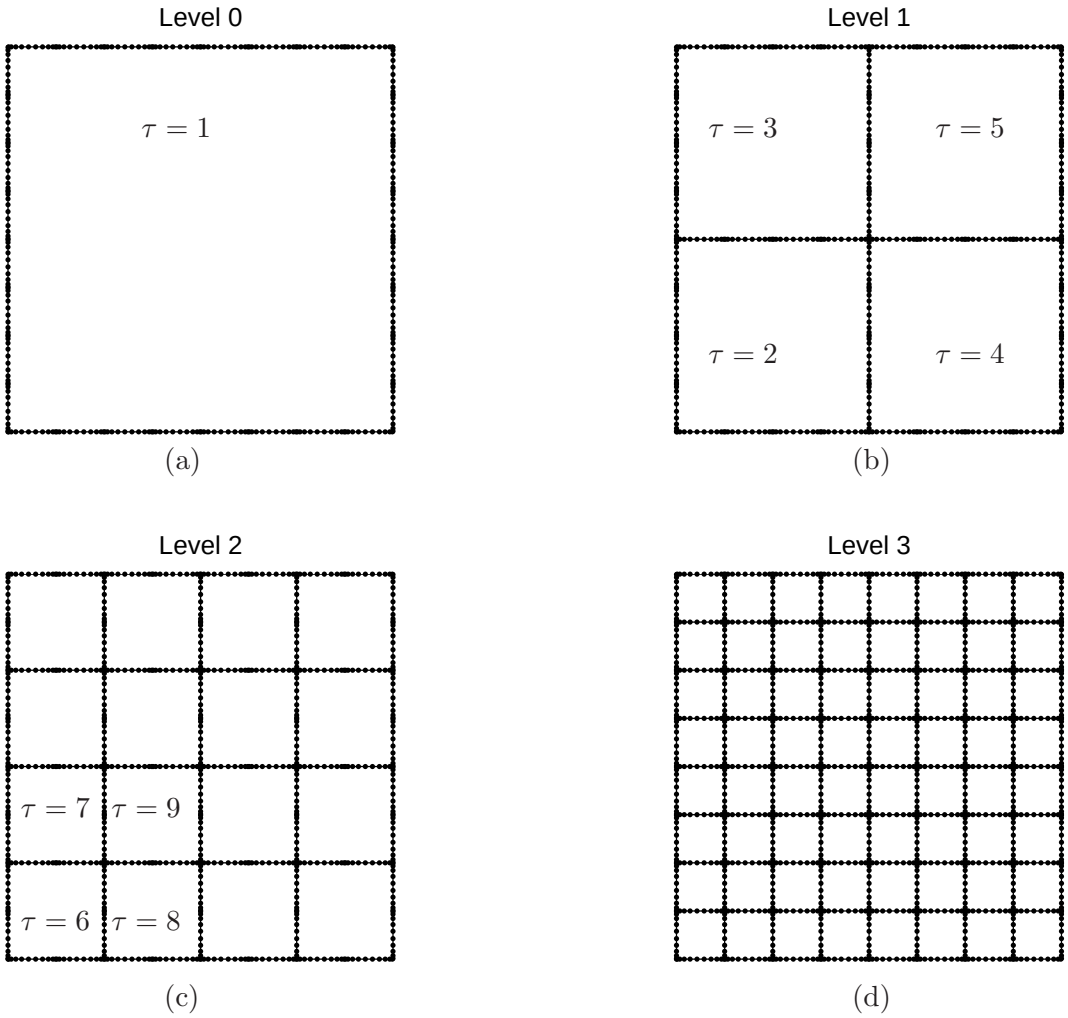


FIGURE 4. Tree structure for a tree with $L = 3$ levels. There are 10 Gaussian nodes on each side of the leaf boxes. The black dots mark the points at which the solution ϕ and its derivative (in the direction normal to the indicated patch boundary) are tabulated.

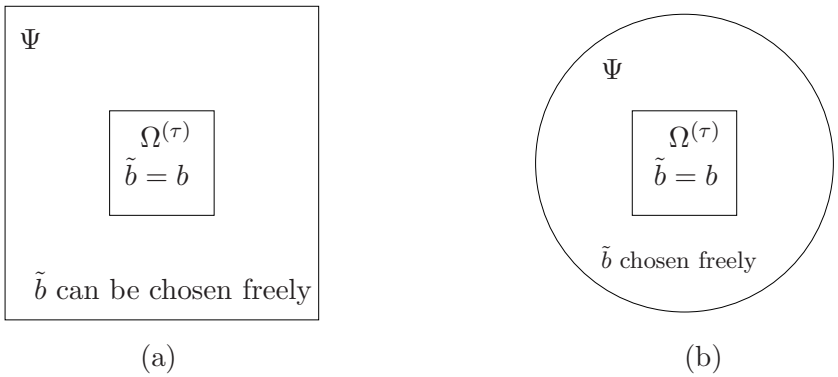


FIGURE 5. Two choices of geometry for the local patch computation. The choice (a) is natural since it conforms to the overall geometry. The advantage of choice (b) is that the FFT can be used in the angular direction.

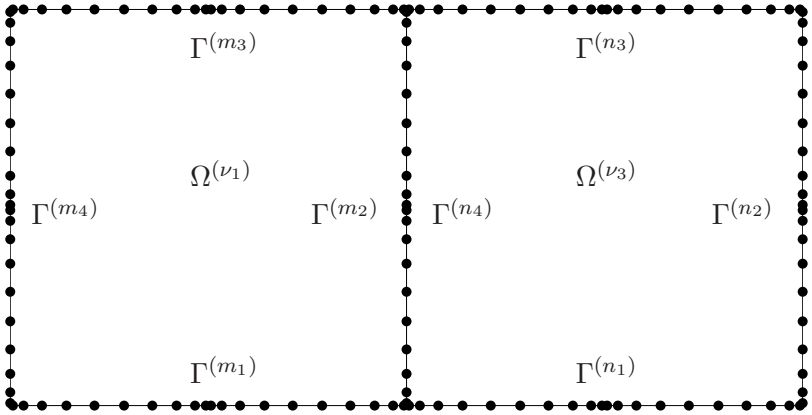


FIGURE 6. Geometry of the *merge* operation.

A high-order accurate discretization scheme for variable coefficient elliptic PDEs in the plane with smooth solutions

P.G. Martinsson, University of Colorado at Boulder, Dec. 23, 2010

1. Introduction

1.1. **Background.** This note describes some tentative ideas for how to discretize and solve a class of variable-coefficient elliptic PDEs with smooth solutions. The ultimate goal is to device efficient methods for (soft) scattering problems in the plane, modeled by the Helmholtz' equation

$$-\Delta\,\phi(x) - \frac{\omega^2}{c(x)^2}\,\phi(x) = 0, \qquad x \in \mathbb{R}^2,$$

where $c(x)$ is a smooth function that is constant outside some domain Ω, and the "loading" of the system is an incoming wave that satisfies the Helmholtz equation for the constant value of c outside Ω.

The method described is high-order accurate and and leads to a linear system of algebraic equations that is very well-suited to "nested-dissection" type *direct* (as opposed to *iterative*) solvers. In a simple implementation, the resulting solver has $O(N^{1.5})$ complexity, where N is the total number of degrees of freedom in the discretization. We believe that the cost of the nested dissection step can be further reduced to $O(N)$ by exploiting techniques similar to those of [ying,grasendyck,xia].

In this initial work, we make several simplifying assumptions in order to investigate the basic viability of the method. The most significant simplification is that instead of studying the Helmholtz equation (which has oscillatory solutions), we study the modified Helmholtz equation (which has non-oscillatory solutions the decay exponentially fast). The remaining simplifications are, we believe, mostly cosmetic.

1.2. **Problem statement.** We consider the equation

$$\begin{cases} -\nabla(a(x)\nabla\phi(x)) + b(x)\,\phi(x) = & 0, \qquad x \in \Omega, \\ \phi_n(x) = & v(x), \qquad x \in \Gamma, \end{cases} \tag{1.1}$$

where Ω is a box in the plane with boundary $\Gamma = \partial\Omega$, and where ϕ_n is the normal derivative of ϕ. We assume that the functions a and b are C^∞, that $b \geq 0$, and that for every $x \in \overline{\Omega}$, $a(x)$ is positive definite. Under these assumptions, the solution ϕ and its gradient $\nabla\phi$ will be C^∞ in the interior of the domain and can to very high accuracy be specified via tabulation at Gaussian quadrature nodes. Near the boundary Γ, the question of smoothness in general gets complicated, but in this preliminary report we sidestep this issue by assuming that the given boundary data v is such that the solution ϕ is C^∞ on the closed domain $\overline{\Omega}$. (Ultimately, the technique will be applied to scattering problems and the function v will be the restriction to Γ of the "incoming wave.")

1.3. **Outline of the discretization scheme.** We propose to tessellate the computational domain into a large number of small squares and then use the fluxes across the boundaries of each square as the unknown variables in the model. The flux is represented as a function along the edge; since this function is smooth, it can very accurately be represented by simply tabulating it at Gaussian nodes along the edge. We observe that if the fluxes through all four edges of a box are known, then the values of the potential ϕ on all of the edges can be constructed via the so called "Neumann-to-Dirichlet" (N2D) operator for the box. These operators can cheaply be constructed for all the boxes via a local computation. Once the N2D operators for all boxes are known, we construct for each edge an equilibrium equation by combining the N2D operators of the two boxes that share the edge. By combining the equilibrium equations for all interior edges, we obtain a global equation for all the interior boundary fluxes. Once this global equation has been solved, the potential on any box can

easily be reconstructed by solving a local Neumann problem on the box (since the boundary fluxes are now all known).

1.4. **Outline of the linear solver.** The discretization described in Section 1.3 results in a large sparse linear system with a coefficient matrix A. If there are N_{edge} edges in the model, and we place N_{gauss} interpolation nodes at each edge, then A is a block matrix consisting of $N_{\text{edge}} \times N_{\text{edge}}$ blocks, each of size $N_{\text{gauss}} \times N_{\text{gauss}}$. By ordering the edges in a nested dissection fashion, fill-in can be limited in the factorization of A, resulting in an $O(N_{\text{edge}}^{1.5})$ total cost for the initial solve. (Once one factorization has been executed, subsequent solves require $O(N_{\text{edge}})$ operations.)

The dominant cost in the factorization described above is the inversion or factorization of dense matrices of size roughly $N_{\text{edge}}^{1/2} \times N_{\text{edge}}^{1/2}$. These matrices have internal structure (they are so call *Hierarchically Semi-Separable* (HSS) matrices) which can be exploited to further reduce the complexity to $O(N_{\text{edge}})$.

2. Discretization

We tessellate Ω into an array of small boxes, and let $\{\Gamma^{(i)}\}_{i \in I_{\text{leaves}}}$ denote the collection of *edges* of these boxes, see Figure 1. We include both interior and exterior edges. For an edge i, we define $u^{(i)}$ as the restriction of ϕ to $\Gamma^{(i)}$:

$$u^{(i)}(x) = \phi(x), \qquad \text{for } x \in \Gamma^{(i)}.$$

Further, we define $v^{(i)}$ as the restriction of the normal derivative across $\Gamma^{(i)}$:

$$v^{(i)}(x) = \begin{cases} [\partial_2 \phi](x) & \text{for } x \in \Gamma^{(i)} \text{ when } \Gamma^{(i)} \text{ is horizontal,} \\ [\partial_1 \phi](x) & \text{for } x \in \Gamma^{(i)} \text{ when } \Gamma^{(i)} \text{ is vertical,} \end{cases}$$

where we used the short-hand $\partial_i = \partial/\partial x_i$.

Observation: All the functions $u^{(i)}$ and $v^{(i)}$ are smooth. They can to very high accuracy be specified by tabulating them at Gaussian points on the boundary and then interpolate between these points.

On each line $\Gamma^{(i)}$, we place N_{gauss} Gaussian nodes. These points are collected in vectors $\boldsymbol{\gamma}^{(i)} \in \mathbb{R}^{N_{\text{gauss}} \times 2}$. Then we form vectors $\boldsymbol{u}^{(i)}, \boldsymbol{v}^{(i)} \in \mathbb{R}^{N_{\text{gauss}}}$ by collocating the boundary functions $u^{(i)}$ and $v^{(i)}$ at the Gaussian nodes:

$$\boldsymbol{u}^{(i)} = u^{(i)}(\boldsymbol{\gamma}^{(i)}),$$
$$\boldsymbol{v}^{(i)} = v^{(i)}(\boldsymbol{\gamma}^{(i)}).$$

3. The equilibrium equations

3.1. **Definition of the Neumann-to-Dirichlet operator.** Let $\Omega^{(\tau)}$ be a subdomain of Ω with edges $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$, as shown in Figure 2. We define the boundary potentials and boundary fluxes for $\Omega^{(\tau)}$ via

$$u^{(\tau)} = \begin{bmatrix} u^{(i_1)} \\ u^{(i_2)} \\ u^{(i_3)} \\ u^{(i_4)} \end{bmatrix} \qquad \text{and} \qquad v^{(\tau)} = \begin{bmatrix} v^{(i_1)} \\ v^{(i_2)} \\ v^{(i_3)} \\ v^{(i_4)} \end{bmatrix}.$$

Then there exists a unique operator $T^{(\tau)}$ such that with $u^{(\tau)}$ and $v^{(\tau)}$ derived from any solution ϕ to (1.1), we have

$$u^{(\tau)} = T^{(\tau)}\, v^{(\tau)}. \tag{3.1}$$

This claim follows from the fact that a Neumann boundary value problem on $\Omega^{(\tau)}$ has a unique solution. The operator $T^{(\tau)}$ is mathematically an integral operator called the *Neumann-to-Dirichlet* operator.

Observation: The N2D operator is in general a complicated object. It has a singular kernel even for domains with smooth boundaries. When the domain boundary has corners, further complications arise. However, in our case, all such subtleties can be ignored since the boundary potentials of interest are all restrictions of functions that globally solve (1.1), and are in consequence smooth.

The discrete analog of the equation (3.1) is

$$\boldsymbol{u}^{(\tau)} = \mathsf{T}^{(\tau)}\,\boldsymbol{v}^{(\tau)}. \tag{3.2}$$

For our purposes, it is sufficient for the matrix $\mathsf{T}^{(\tau)}$ to correctly construct $\boldsymbol{u}^{(\tau)}$ for any *permissible* vectors $\boldsymbol{v}^{(\tau)}$. (By permissible, we mean that they are the restriction of a function in the solution set under consideration.) Written out in components, $\mathsf{T}^{(\tau)}$ is a 4×4 block matrix that satisfies

$$\begin{bmatrix} \boldsymbol{u}^{(i_1)} \\ \boldsymbol{u}^{(i_2)} \\ \boldsymbol{u}^{(i_3)} \\ \boldsymbol{u}^{(i_4)} \end{bmatrix} = \begin{bmatrix} \mathsf{T}^{(\tau,11)} & \mathsf{T}^{(\tau,12)} & \mathsf{T}^{(\tau,13)} & \mathsf{T}^{(\tau,14)} \\ \mathsf{T}^{(\tau,21)} & \mathsf{T}^{(\tau,22)} & \mathsf{T}^{(\tau,23)} & \mathsf{T}^{(\tau,24)} \\ \mathsf{T}^{(\tau,31)} & \mathsf{T}^{(\tau,32)} & \mathsf{T}^{(\tau,33)} & \mathsf{T}^{(\tau,34)} \\ \mathsf{T}^{(\tau,41)} & \mathsf{T}^{(\tau,42)} & \mathsf{T}^{(\tau,43)} & \mathsf{T}^{(\tau,44)} \end{bmatrix} \begin{bmatrix} \boldsymbol{v}^{(i_1)} \\ \boldsymbol{v}^{(i_2)} \\ \boldsymbol{v}^{(i_3)} \\ \boldsymbol{v}^{(i_4)} \end{bmatrix}. \tag{3.3}$$

3.2. **Construction of the N2D operator for a small box.** Let $\Omega^{(\tau)}$ be a small box with edges $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$, as shown in Figure 2, and consider the task of constructing a matrix $\mathsf{T}^{(\tau)}$ such that (3.2) holds for all permissible potentials. We generate (by brute force) a collection of solutions $\{\phi_j\}_{j=1}^{N_{\text{samp}}}$ that locally span the solution space to the desired precision. For each ϕ_j, we construct the corresponding vectors of boundary values

$$\boldsymbol{u}_j = \begin{bmatrix} \phi_j(\boldsymbol{\gamma}^{(i_1)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_2)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_3)}) \\ \phi_j(\boldsymbol{\gamma}^{(i_4)}) \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{v}_j = \begin{bmatrix} \partial_2\phi_j(\boldsymbol{\gamma}^{(i_1)}) \\ \partial_1\phi_j(\boldsymbol{\gamma}^{(i_2)}) \\ \partial_2\phi_j(\boldsymbol{\gamma}^{(i_3)}) \\ \partial_1\phi_j(\boldsymbol{\gamma}^{(i_4)}) \end{bmatrix}$$

and then we construct via a least squares procedure a matrix $\mathsf{T}^{(\tau)}$ such that the equation

$$[\boldsymbol{u}_1\ \boldsymbol{u}_2\ \cdots\ \boldsymbol{u}_{N_{\text{samp}}}] = \mathsf{T}^{(\tau)}\,[\boldsymbol{v}_1\ \boldsymbol{v}_2\ \cdots\ \boldsymbol{v}_{N_{\text{samp}}}] \tag{3.4}$$

holds to within the specified tolerance ε.

The sample functions ϕ_j are constructed by solving a set of local problems on a patch Ψ that covers the domain $\Omega^{(\tau)}$, as shown in Figure 5. The local problems read

$$\begin{cases} -\nabla(\tilde{a}(x)\nabla\phi_j(x)) + \tilde{b}(x)\,\phi_j(x) = & 0, & x \in \Psi, \\ \partial_n\phi(x) = & v_j(x), & x \in \partial\Psi, \end{cases} \tag{3.5}$$

where $\tilde{a}$ and $\tilde{b}$ are functions chosen so that:

(1) For $x \in \Omega^{(\tau)}$, we have $\tilde{a}(x) = a(x)$ and $\tilde{b}(x) = b(x)$.
(2) The equation (3.5) is easy to solve.

The collection of boundary data $\{v_j\}_{j=1}^{N_{\text{samp}}}$ is chosen so that the solution space is sufficiently "rich."

3.3. **Assembling a global equilibrium equation.** In this section, we will formulate a linear equation that relates the following variables:

$$\begin{aligned} \text{Given data:} \quad & \{\boldsymbol{v}^{(i)} : i \text{ is an edge that is exterior to } \Omega\}, \\ \text{Sought data:} \quad & \{\boldsymbol{v}^{(i)} : i \text{ is an edge that is interior to } \Omega\}. \end{aligned}$$

Let N_{edge} denote the number of interior edges. Then the coefficient matrix of the linear system will consist of $N_{\text{edge}} \times N_{\text{edge}}$ blocks, each of size $N_{\text{gauss}} \times N_{\text{gauss}}$. Each block row in the system will have at most 7 non-zero blocks. To form this matrix, let i denote an interior edge. Suppose for a moment that i is a vertical edge. Let τ_1 and τ_2 denote the two boxes that share the edge i, let $\{m_1, m_2, m_3, m_4\}$ denote the edges of τ_1, and let $\{n_1, n_2, n_3, n_4\}$ denote the edges of τ_2, see Figure 3. The N2D operator for τ_1 provides an equation for the boundary fluxes of the left box:

$$\boldsymbol{u}^{(m_2)} = \mathsf{T}^{(\tau_1,21)}\,\boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\tau_1,22)}\,\boldsymbol{v}^{(m_2)} + \mathsf{T}^{(\tau_1,23)}\,\boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\tau_1,24)}\,\boldsymbol{v}^{(m_4)}. \tag{3.6}$$

Analogously, the N2D operator for τ_2 provides the equation

$$\boldsymbol{u}^{(n_4)} = \mathsf{T}^{(\tau_2,41)}\,\boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\tau_2,42)}\,\boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\tau_2,43)}\,\boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\tau_2,44)}\,\boldsymbol{v}^{(n_4)}. \tag{3.7}$$

Observing that $m_2 = n_2 = i$, we see that $\boldsymbol{u}^{(m_2)} = \boldsymbol{u}^{(n_4)}$, and consequently (3.6) and (3.7) can be combined to form the equation

$$\begin{aligned}\mathsf{T}^{(\tau_1,21)}\,\boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\tau_1,22)}\,\boldsymbol{v}^{(i)} + \mathsf{T}^{(\tau_1,23)}\,\boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\tau_1,24)}\,\boldsymbol{v}^{(m_4)} \\ = \mathsf{T}^{(\tau_2,41)}\,\boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\tau_2,42)}\,\boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\tau_2,43)}\,\boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\tau_2,44)}\,\boldsymbol{v}^{(i)}.\end{aligned} \tag{3.8}$$

The collection of all equations of the form (3.8) for interior vertical edges, along with the analogous set of equations for all interior horizontal edges forms the global equilibrium equation.

4. Efficient direct solvers

This section describes a direct solver for the global equilibrium equation constructed in Section 3. The idea is to partition the box Ω into a quad-tree of boxes, and then to construct the N2D operator $\mathsf{T}^{(\tau)}$ for each box τ in the tree. The first step is to loop over all leaf nodes of the tree and construct the N2D operator via the procedure described in Section 3.2. Then we execute an upwards sweep through the tree, where we construct the N2D operator for a box by merging the operators of its four children. For simplicity (and also computational efficiency) we execute each "merge-four" operation as a set of three "merge-two" operations.

Let N denote the size of the coefficient matrix. Then Section 4.2 describes a procedure with $O(N^{1.5})$ complexity, and Section 4.3 sketches out how the procedure can be accelerated to $O(N)$ complexity. Before describing the fast solvers, we describe a hierarchical decomposition of the domain in Section 4.1.

4.1. A quad-tree on the domain.

A standard quad-tree is formed on the computational domain Ω as follows: Let $\Omega^{(1)} = \Omega$ be the *root* of the tree, as shown in Figure 4(a). Then split $\Omega^{(1)}$ into four disjoint boxes

$$\Omega^{(1)} = \Omega^{(2)} \cup \Omega^{(3)} \cup \Omega^{(4)} \cup \Omega^{(5)},$$

as shown in Figure 4(b). Continue by splitting each of the four boxes into four smaller equisized boxes:

$$\begin{aligned}\Omega^{(5)} &= \Omega^{(6)} \cup \Omega^{(7)} \cup \Omega^{(8)} \cup \Omega^{(9)},\\ \Omega^{(6)} &= \Omega^{(10)} \cup \Omega^{(11)} \cup \Omega^{(12)} \cup \Omega^{(13)},\\ \Omega^{(7)} &= \Omega^{(14)} \cup \Omega^{(15)} \cup \Omega^{(16)} \cup \Omega^{(17)},\\ \Omega^{(8)} &= \Omega^{(18)} \cup \Omega^{(19)} \cup \Omega^{(20)} \cup \Omega^{(21)},\end{aligned}$$

as shown in Figure 4(c). The process continues until each box is small enough that the N2D operator for each leaf can easily be constructed via the procedure described in Section 3.2. The levels of the tree are ordered so that $\ell = 0$ is the coarsest level (consisting only of the root), $\ell = 1$ is the level with four boxes, etc. We let L denote the total number of levels in the tree.

4.2. **Simple construction of the N2D operator for a parent.** Suppose that σ is a box with children ν_1 and ν_3 as shown in Figure 6, and that we know the matrices $\mathsf{T}^{(\nu_1)}$ and $\mathsf{T}^{(\nu_3)}$ associated with the children. We seek to construct the matrix $\mathsf{T}^{(\sigma)}$. The equilibrium equations for the two children read

$$\boldsymbol{u}^{(m_i)} = \sum_{j=1}^{4} \mathsf{T}^{(\nu_1,ij)}\, \boldsymbol{v}^{(m_j)}, \qquad i = 1,\, 2,\, 3,\, 4, \tag{4.1}$$

$$\boldsymbol{u}^{(n_i)} = \sum_{j=1}^{4} \mathsf{T}^{(\nu_3,ij)}\, \boldsymbol{v}^{(n_j)}, \qquad i = 1,\, 2,\, 3,\, 4. \tag{4.2}$$

Observing that $\boldsymbol{u}^{(m_2)} = \boldsymbol{u}^{(n_n)}$ we combine (4.1) for $i = 2$ with (4.2) for $i = 4$ to obtain the joint equation

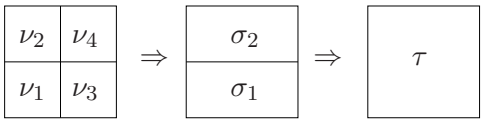
$$\begin{aligned}\mathsf{T}^{(\nu_1,21)}\, \boldsymbol{v}^{(m_1)} + \mathsf{T}^{(\nu_1,22)}\, \boldsymbol{v}^{(m_2)} + \mathsf{T}^{(\nu_1,23)}\, \boldsymbol{v}^{(m_3)} + \mathsf{T}^{(\nu_1,24)}\, \boldsymbol{v}^{(m_4)} \\ = \mathsf{T}^{(\nu_3,41)}\, \boldsymbol{v}^{(n_1)} + \mathsf{T}^{(\nu_3,42)}\, \boldsymbol{v}^{(n_2)} + \mathsf{T}^{(\nu_3,43)}\, \boldsymbol{v}^{(n_3)} + \mathsf{T}^{(\nu_3,44)}\, \boldsymbol{v}^{(n_4)}.\end{aligned} \tag{4.3}$$

Utilizing further that $\boldsymbol{v}^{(m_2)} = \boldsymbol{v}^{(n_4)}$, we write (4.3) along with (4.1) and (4.2) as

$$\left[\begin{array}{ccc|ccc|c} \mathsf{T}^{(\nu_1,11)} & \mathsf{T}^{(\nu_1,13)} & \mathsf{T}^{(\nu_1,14)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,12)} \\ \mathsf{T}^{(\nu_1,31)} & \mathsf{T}^{(\nu_1,33)} & \mathsf{T}^{(\nu_1,34)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,32)} \\ \mathsf{T}^{(\nu_1,41)} & \mathsf{T}^{(\nu_1,43)} & \mathsf{T}^{(\nu_1,44)} & 0 & 0 & 0 & \mathsf{T}^{(\nu_1,42)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,11)} & \mathsf{T}^{(\nu_3,12)} & \mathsf{T}^{(\nu_3,13)} & \mathsf{T}^{(\nu_3,14)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,21)} & \mathsf{T}^{(\nu_3,22)} & \mathsf{T}^{(\nu_3,23)} & \mathsf{T}^{(\nu_3,24)} \\ 0 & 0 & 0 & \mathsf{T}^{(\nu_3,31)} & \mathsf{T}^{(\nu_3,32)} & \mathsf{T}^{(\nu_3,33)} & \mathsf{T}^{(\nu_3,34)} \\ \hline \mathsf{T}^{(\nu_1,21)} & \mathsf{T}^{(\nu_1,23)} & \mathsf{T}^{(\nu_1,24)} & -\mathsf{T}^{(\nu_3,41)} & -\mathsf{T}^{(\nu_3,42)} & -\mathsf{T}^{(\nu_3,43)} & \mathsf{T}^{(\nu_1,22)} - \mathsf{T}^{(\nu_3,44)} \end{array}\right] \left[\begin{array}{c} \boldsymbol{v}^{(m_1)} \\ \boldsymbol{v}^{(m_3)} \\ \boldsymbol{v}^{(m_4)} \\ \boldsymbol{v}^{(n_1)} \\ \boldsymbol{v}^{(n_2)} \\ \boldsymbol{v}^{(n_3)} \\ \hline \boldsymbol{v}^{(m_2)} \end{array}\right] = \left[\begin{array}{c} \boldsymbol{u}^{(m_1)} \\ \boldsymbol{u}^{(m_3)} \\ \boldsymbol{u}^{(m_4)} \\ \boldsymbol{u}^{(n_1)} \\ \boldsymbol{u}^{(n_2)} \\ \boldsymbol{u}^{(n_3)} \\ \hline \boldsymbol{0} \end{array}\right].$$

Eliminating $\boldsymbol{v}^{(m_2)}$ from the system via a Schur complement yields the operator $\mathsf{T}^{(\sigma)}$ (upon suitable reblocking).

At this point, we have described a "merge-two" operation. A "merge-four" operation can of course be obtained by simply combining three merge-two operations. To be precise, suppose that τ is a node with the four children ν_1, ν_2, ν_3, ν_4. We introduce the two "intermediate" boxes σ_1 and σ_2 as shown in the following figure:



Letting the procedure described earlier in the section be denoted by "`merge_two_horizontal`" and defining an analogous function "`merge_two_vertical`" we then find that

$$\begin{aligned}\mathsf{T}^{(\sigma_1)} &= \texttt{merge\_two\_horizontal}(\mathsf{T}^{(\nu_1)},\, \mathsf{T}^{(\nu_3)}),\\ \mathsf{T}^{(\sigma_2)} &= \texttt{merge\_two\_horizontal}(\mathsf{T}^{(\nu_2)},\, \mathsf{T}^{(\nu_4)}),\\ \mathsf{T}^{(\tau)} &= \texttt{merge\_two\_vertical}(\mathsf{T}^{(\sigma_1)},\, \mathsf{T}^{(\sigma_2)}).\end{aligned}$$

4.3. **Fast construction of the N2D operator for a parent.** The merge operation described in Section 4.2 has asymptotic cost $O(N^{1.5})$, where N is the total number of points on the edges of the leaves. To simplify slightly, the reason is that forming the "merge" operation requires matrix inversion and matrix-matrix-multiplications for of dense matrices whose size eventually grow to $O(\sqrt{N}) \times O(\sqrt{N})$. However, these matrices all have internal structure. To be precise, in (3.3), the off-diagonal blocks have low rank, and the diagonal blocks are all Hierarchically Semi-Separable (HSS) matrices. This means that accelerated matrix algebra can be used. For a matrix of size $N' \times N'$, inversion can in fact be executed in $O(N')$ operations (provided that the "HSS-rank" is a fixed low number, which it is in this case).

The acceleration procedure proposed here is analogous to the one suggested in [Xia,Ying].

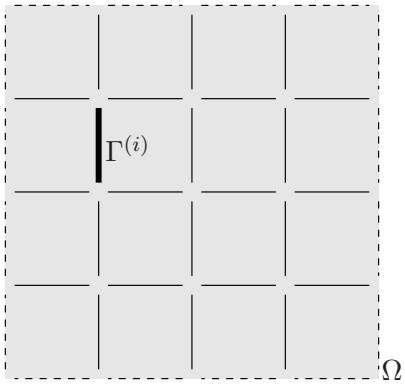


FIGURE 1. The computational box Ω (gray) is split into 16 small boxes. There are a total of 40 edges in the discretization, 24 interior ones (solid lines) and 16 exterior ones (dashed lines). One interior edge $\Gamma^{(i)}$ is marked with a bold line. (Each edge continues all the way to the corner, but has been drawn slightly forshortened for clarity.)

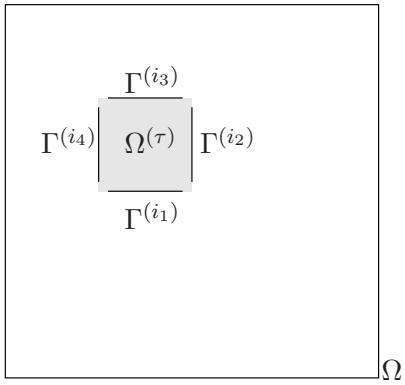


FIGURE 2. The box $\Omega^{(\tau)}$ is marked in gray. Its edges are $\Gamma^{(i_1)}$, $\Gamma^{(i_2)}$, $\Gamma^{(i_3)}$, $\Gamma^{(i_4)}$.

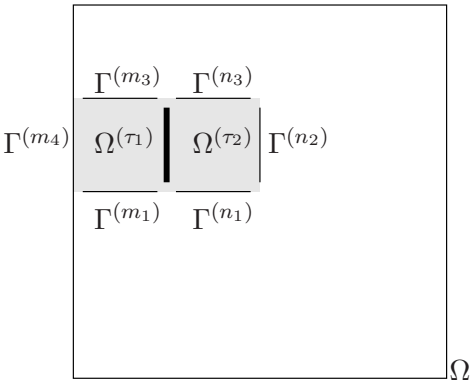


FIGURE 3. Construction of the equilibrium equation for the edge $\Gamma^{(i)}$ in Figure 1. It is the common edge of the boxes $\Omega^{(\tau_1)}$ and $\Omega^{(\tau_2)}$, which have edges $\{\Gamma^{(m_1)}, \Gamma^{(m_2)}, \Gamma^{(m_3)}, \Gamma^{(m_4)}\}$, and $\{\Gamma^{(n_1)}, \Gamma^{(n_2)}, \Gamma^{(n_3)}, \Gamma^{(n_4)}\}$, respectively. Observe that $\Gamma^{(i)} = \Gamma^{(m_2)} = \Gamma^{(n_4)}$ (the bold line).

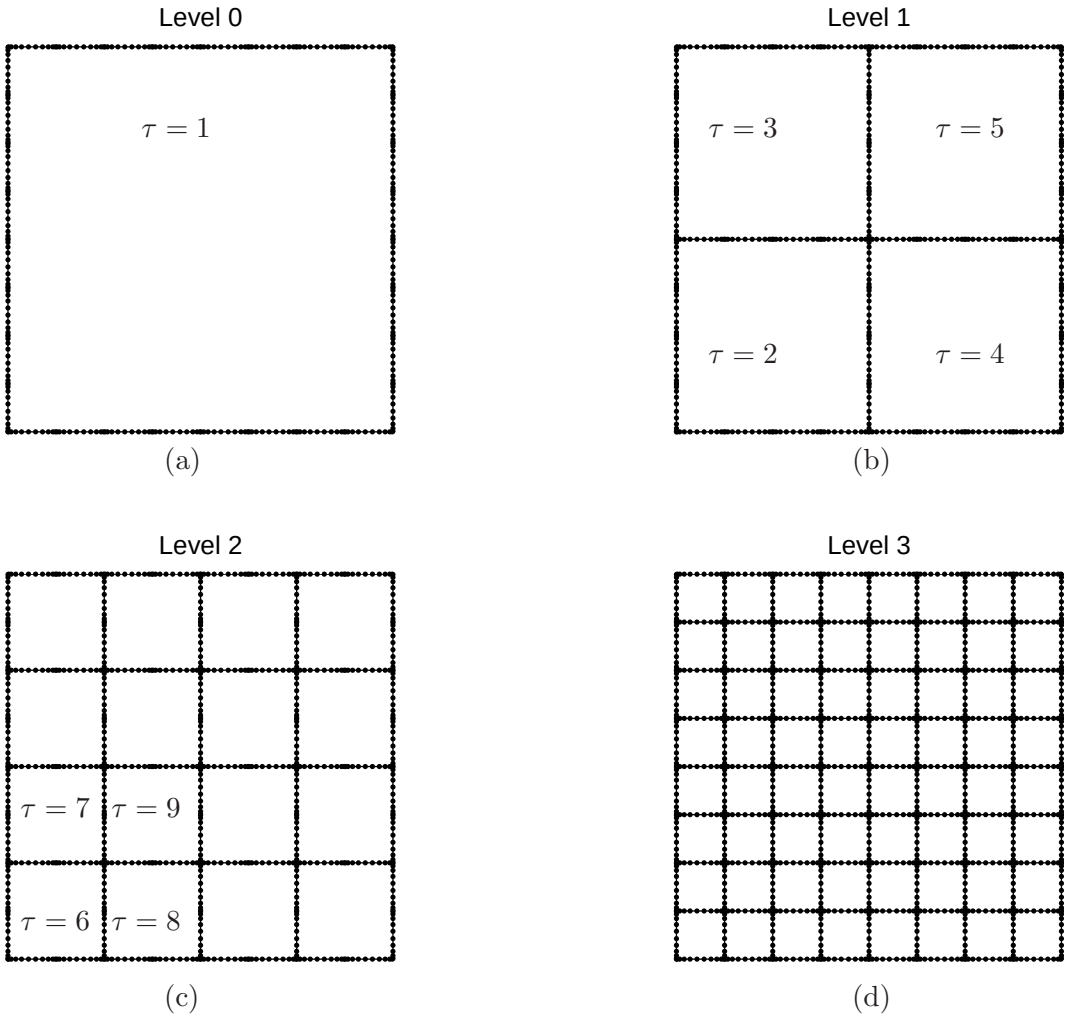


FIGURE 4. Tree structure for a tree with $L = 3$ levels. There are 10 Gaussian nodes on each side of the leaf boxes. The black dots mark the points at which the solution ϕ and its derivative (in the direction normal to the indicated patch boundary) are tabulated.

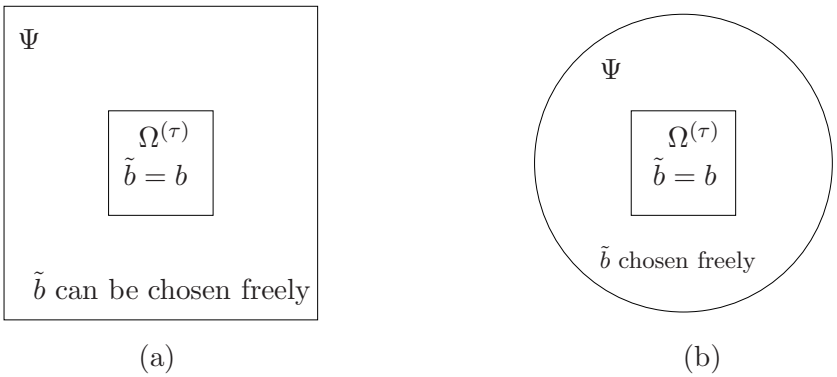


FIGURE 5. Two choices of geometry for the local patch computation. The choice (a) is natural since it conforms to the overall geometry. The advantage of choice (b) is that the FFT can be used in the angular direction.

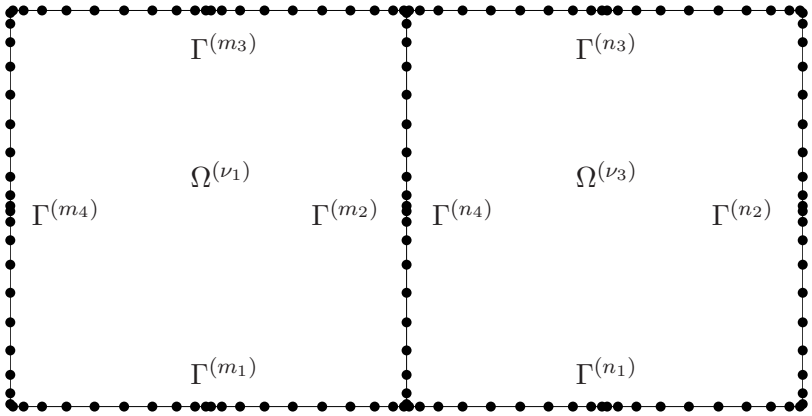


FIGURE 6. Geometry of the *merge* operation.

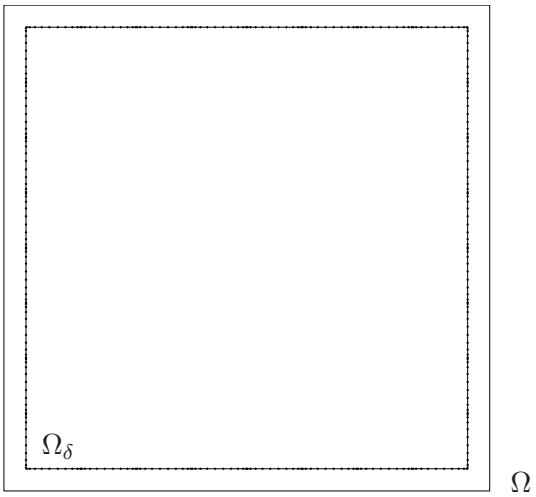


FIGURE 7. Geometry of the method described in Section **??**. The strip between the two squares has thickness δ.